Implementation of prioritised task scheduling using resource optimization based on sub schedulers

Toofan Mukherjee , ShrinathTailor , Leela Kumari

Abstract: The utilisation of computation, software, and platform infrastructure as a service through the Internet is referred to as cloud computing. Customers do not own or they still manage their entire computer resources, paying only for what they use. In this case, the job scheduling algorithm may not only match user needs, but also achieve high resource utilisation, improved performance, and faster reaction times. [3]. The cloud delivers scalable virtual machines where work can be distributed, but scheduling is a major challenge. Previous tasks used parameters to complete task prioritization. Scholars specify the length of a task and then schedule it using max and min methods, but the problem is that prioritizing tasks based on length alone is not ideal. To fill in the blanks of previous work. ,the researchers suggested sorting task priorities based on four factors: task length, estimated cost of the task, due date, and task priority. The proposed task scheduling was correct based on these four variables. Second, it predicts the execution and completion times of all tasks for all available resources, and accurately associates virtual machines and tasks to achieve resource awareness, which increases reliability over traditional performance.

Keywords : cloud computing,task scheduling Cloud-sim, MIN-MAX algorithm.

I.INTRODUCTION

Cloud computing is progressing at a rapid pace. It also allows users to access apps and related data from any location. A collection of computing and communication resources deployed across several datacenters and used by a diverse set of users is referred to as CC [7]. The data centre network framework scalability, energy preservation, duplicate policies, safety, as well as scheduling method are some of the disadvantages of cloud computing[2]. The chief objective of this paper is to optimize scheduling policies. A fundamental tenet is a task scheduling approach. To introduce a set of cloud services in order to ensure an effective provider infrastructure. The task scheduling strategy is in responsibility of mapping workloads sent to the cloud environment to resources that are available in order to minimize total response time, or the make span. Numerous tasks in cloud computing must be completed by on-hand resources in order to achieve the quickest total time to complete, the quickest responsiveness, and the most effective resource use [3], and so on. Because of these disparities, researchers need to develop, improve, and suggested a scheduling approach that task schedulers can use to create the best suitable allocation map of tasks on resources.

There are numerous methods that are used to develop a program on their resources. Max-min, Min-min, and RASA [4] are three well-known examples of such techniques intended at being applied in a CC setting. Each of these approaches forecasts the completion and execution times of each task offered on each tool. RASA is a combination of two different algorithms. In the RASA, an estimate of the task's completion time is computed based on available resources, and then the Max-min and Min-min approaches are alternated to reap the benefits of both while avoiding their drawbacks.

The article is structured as obeyed: Section II explains Task Scheduling. Section III & IV Scheduling of tasks and Literature survey. Section V explained the proposed methodology .Section VI explains results. At end, conclusion is in Section VII.

II. TASK SCHEDULING

The method of allocating one or more time periods to one or more resources is known as task scheduling. It's difficult to organise a group of submitted tasks from numerous users on a set of computer resources in a cloud environment to reduce mission completion time or system make span. Scheduling is a collection of policies that govern the order in which a computer system performs its tasks. In distributed computing systems, different kinds of scheduling algorithms exist. The key advantage of scheduling algorithms is that they achieve high computational as well as the best throughput [3].

Scheduling manages CPU memory accessibility, as well as a good scheduling policy maximizes resource utilization. Scheduling algorithms can be divided into several policies, Instant and batch planning, preemptive and non-preemptive organising, static and dynamic scheduling are all examples[5]. In immediate mode, activities are placed in computing environment as rapidly as possible, although in batch mode, the first task are gathered into a batch i. e. a collection of meta-tasks might be issued at mapping events. Jobs are pre-scheduled in Static Scheduling, all information about available finances and operations is known ahead of time, and a job is assigned to a source just once. In Dynamic Scheduling, tasks are dynamically available for scheduling by planner. The ability to determine run time in advance is more flexible than static scheduling. In Pre-emptive, there is a task for every task. During execution, scheduling may be disrupted, and a job may be relocated to another resource while the original assigned resource remains available for other activities. This scheduling technique becomes more useful when limitations such as priority are taken into account [6]. Resources cannot be re-allocated in the Non Pre-emptive Scheduling scheme until the running as well as scheduled job has completed its execution.

III. Scheduling of Tasks

The primary goal of task scheduling approaches in Cloud computing devices is to distribute the load on processors while maximizing resource utilization and minimizing make span. In CC, task scheduling could be divided into three stages:

Resource detection & filtering: The broker chooses the available resources in the cloud computing platform and collects status information about them.

Resource allocation: The necessary resource is chosen based on the parameters of the task as well as the resource.

Task submission: The priority is performed to the appropriate resource.

Min-min as well as max-min methodologies are modeled to reduce the overall duration of tasks on machines and present better performance. To obtain the shortest makespan, these techniques are modeled in both space-shared and time-shared modes.

The Min-Min[7] algorithm works as follows:

Phase 1: First, it determines how long each task on each machine takes to complete, and then it calculates the average time for each task. It selects the machine that analyses the tasks in the shortest amount of time for each task.

Phase 2: The job with shortest completion time is selected and assigned to machine with least execution time among all jobs in Meta task. The task is removed from the Meta Task list, and the cycle repeats till the criterion is met.

The Max-Min algorithm operates as follows:

Phase 1: evaluates the completion time of each task on every machine, then selects the machine that analyses the tasks in the shortest amount of time for each task.

Phase 2: The job with the shortest completion time is picked and recognised to the machine from all the tasks in Meta Task. The work is removed from the Meta Task list, and the step was repeated till the queue is empty [7].

To properly appreciate how these scheduling algorithms work in order to attain the shortest possible time in a CC environment, a simulated environment was created using the CloudSim software application. Events are presented as Cloudlets, computers as VMs, and cloud itself as Datacenter in CloudSim. Cloudlets are planned on virtual machines (VMs), with each VM being a virtual machine that cloudlets execute on.

IV.LITERATURE SURVEY

Amalarethinam et al.,[8] Priority Based Performance Improved Method is suggested, that considers the user's preference. Users who use cloud services frequently may experience good service. Regular user tasks are, of course, given top priority. Many activities are of regular priority. The Meta feature set should be categorized into two parts: important tasks as well as low priority tasks. The Using the Min-Min technique, the activities of the task group with the highest priority should be scheduled first. Then, using the Max-Min approach, schedule the key tasks that need to be done. This method, like the Min-Min method, determines the average time for priority jobs. When compared to the Min-min approach, this method is more efficient, Make span decrease it while increasing resource utilization.

T.Mandal et al.,[9] mentioned meta-heuristic strategies for optimal task scheduling in a cloud computing environment. CC is the most recent development of grid computing, distributed computing, as well as parallel computing. In this scheme, users can access various services such as storage, servers, or other apps. Cloud resources are not only used by a large number of users, but they are also redistributed arbitrarily on demand. The Internet is used to deliver requested services to users' devices. To identify the optimum, three meta-heuristic methods were used in this work: Simulated Annealing, FA, as well as CS Algorithm. The main goal of these strategies is to decrease the overall processing time of virtual machines that perform a set of tasks. The results of experiments reveal that FFA outperforms both SA and CS Algorithm.

Derakhshan et al.,[10] presented an optimal method by combining the benefits of Min-Min as well as Max-Min methodologies. The main focus of the tasks is another point that follows in this method (TOMMP). There are many scheduling approaches in use today, However, in the majority of systems, task prioritisation has been disregarded or overlooked. A prioritising strategy is used to assign priority to jobs, and the median number is utilised to determine whether the Min-Min or Max-Min approach should be used. It should be mentioned that the TOMMP algorithms have a shorter waiting time than the compared algorithms and are proved to be superior to the comparable techniques.

Ibrahim et al.,[11] An improved task scheduling policy is suggested, in which obtainable VMs are assigned to needing tasks based on their processing power as well as execution cost. The researchers used Amazon EC2 as well as Google pricing schemes to effectively estimate as well as calculate the implementation cost/price. This system could be improved further by incorporating dynamic workflow scheduling as well as dependent tasks.

Santhosh et al.,[12] For cloud computing, a developed max-min scheduling technique is designed. This is a fix for the advanced Max-min method. It calculates the average execution speed of all available jobs first, then determines the run-time distance that is closest to the average. Frequently, the largest job is overly large, resulting in a system imbalance. As a result, this method always chooses the task whose duration or execution time is nearest to the average length or execution time of all jobs. Over the Maxmin technique, this approach minimizes Make span and accomplishes load balancing.

Jain et al.,[13] Expense, power, resource consumption, rotating time, and duration can all be improved by using optimization. To decrease Make span as well as increase resource productivity, a computational resource was suggested. This method schedules huge set of different jobs for evaluating Make span and CPU variables using particle optimization integration as well as the Max-Min method. As a result, this method utilizes a combination of both methodologies while maintaining high accuracy. It takes a long time for the traditional PSO algorithm for tracing best way. However, it creates output for potential job sequencing that could be assigned to a virtual machine when combined with the Max-Min approach. The PSO algorithm is then fed this output. As a consequence, it allows for the best solution to be found quicker than the traditional one. The testing findings show that this suggested algorithm outperforms other methodologies.

V.PROPOSED METHODOLOGY

- **Problem Formulation**

There are several scheduling methods available that are used in the cloud data centers for improved resource planning. Scheduling entails allocating a pool of assets to many tasks in such a way that maximal resource usage, total process time and waiting period are all kept upto minimum extent. Conventional or regulation based scheduling algorithms are commonly used it on cloud computing environment since they are easy to execute, and they can provide enough an optimum solutions for scheduling issues. However, they all suffer from the effects of being not allowed to find the best possible solution in a timely manner for too complicated or big problems. Job scheduling is a procedure of mapping from tasks of users to the suitable resources' selection and its execution also.

The work in the existing paper is based on task prioritisation utilising some credits based on job duration, followed by scheduling using the max-min and min-min algorithms, and the outcomes are assessed. Allocating priority solely on the basis of task length will result in two or more tasks with same priority, making it difficult to assign tasks based on priority. As a result, an algorithm is proposed that will address the issues raised by this method by suggesting a methodology based on a larger number of credits, which includes task length, prioritised divisible factor, expense, and time limit. The tasks are then prioritised, and the idea of asset aware predicated on fill and spill sub-scheduler is suggested to utilize human resources.

Research Objectives

The objectives of this study are as follows:

- To study and analyze the existing scheduling algorithms in cloud computing
- To implement an improvised, prioritize task scheduling algorithm with resource optimization based on sub-schedulers.
- To evaluate the performance of improvised algorithm based on various scenarios of jobs and machines and compare it with existing base techniques on the basis total processing time, total processing cost and makespan time.

- **Methodology Used**

The proposed algorithm employs the notion of work prioritising based on a larger number of credits, which include task duration, cost, deadline, and priority, as well as managing the solution's resources. By evaluating the demand and processing capacity on each virtual machine, the resource aware load balancing method ensures system load balancing. It consists of two phases: scheduling based on VM computing capabilities, and selecting the VM with the earliest less load for jobs mapping, which involves transferring extra tasks from an overloaded VM to an underloaded VM based on demand and supply numbers. Resource aware optimization method is used in the proposed load balancing algorithm to balance load among VMs and manage resources. The Fill and Spill schedulers are two sub-schedulers of RALBA. Fill Scheduler allocates Cloudlets to VMs based on computational capacity of VMs. Fill scheduler chooses VMj with highest VM Share and decides VMj's maximum PCloudletVMj. Cloudlet to VM allocation is conducted, and VMSharej of VMj is modified after Cloudlet is allocated. Fill scheduling repeats cloudlets mapping procedure till no VMj with non-empty RPCloudletj exists or CLS is emptied. Spill scheduler assigns Cloudlets to VMs depending on EFT of candidate Cloudlet. RALBA system moves to Spill scheduler after the Fill scheduler to assign leftover CLS Cloudlets. maxCloudlet is chosen and assigned to VM that generates EFT for this maxCloudlet. VM's time duration is upgraded on contender Cloudlet–VM assignment. The assignment of Cloudlets to VMs is repeated till CLS is clear. Outcomes are then optimised and jobs are scheduled based on this vm allocation list.

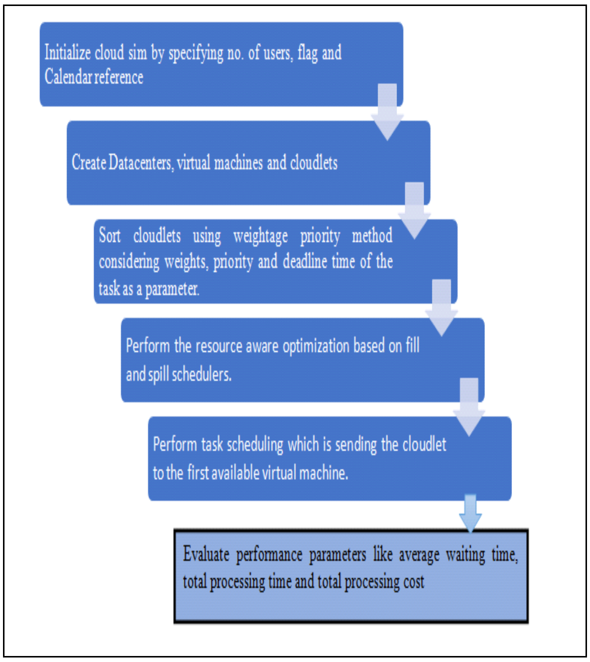
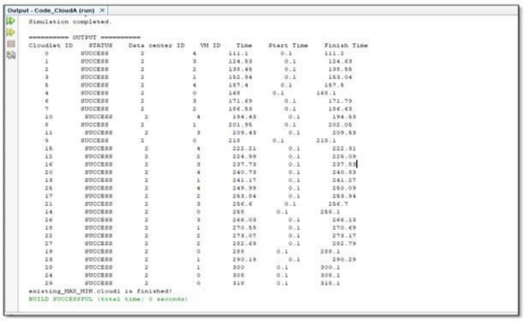


Figure 1: Flowchart of the proposed approach

VI. RESULTS

The methodology introduced here has been evaluated in a simulated cloud environment. The "CloudSim" simulator is being used for simulation. The completion time as well as average waiting time of tasks are compared using two methods, where the first system's scheduling requirement is only task length and the second computation scheduling standard is only task deadline. The proposed computation outcomes have been enhanced.



Output - Code_CloudA (run)

Simulation completed.

========== OUTPUT ==========

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	4	111.1	0.1	111.2
1	SUCCESS	2	3	124.53	0.1	124.63
2	SUCCESS	2	2	138.45	0.1	138.55
3	SUCCESS	2	1	152.94	0.1	153.04
5	SUCCESS	2	4	157.4	0.1	157.5
4	SUCCESS	2	0	168	0.1	168.1
6	SUCCESS	2	3	171.69	0.1	171.79
7	SUCCESS	2	2	186.53	0.1	186.63
10	SUCCESS	2	4	194.43	0.1	194.53
8	SUCCESS	2	1	201.95	0.1	202.05
11	SUCCESS	2	3	209.43	0.1	209.53
9	SUCCESS	2	0	210	0.1	210.1
15	SUCCESS	2	4	222.21	0.1	222.31
12	SUCCESS	2	2	224.99	0.1	225.09
16	SUCCESS	2	3	237.73	0.1	237.83
20	SUCCESS	2	4	240.73	0.1	240.83
13	SUCCESS	2	1	241.17	0.1	241.27
25	SUCCESS	2	4	249.99	0.1	250.09
17	SUCCESS	2	2	253.84	0.1	253.94
21	SUCCESS	2	3	256.6	0.1	256.7
14	SUCCESS	2	0	258	0.1	258.1
26	SUCCESS	2	3	266.03	0.1	266.13
18	SUCCESS	2	1	270.59	0.1	270.69
22	SUCCESS	2	2	273.07	0.1	273.17
27	SUCCESS	2	2	282.69	0.1	282.79
19	SUCCESS	2	0	288	0.1	288.1
23	SUCCESS	2	1	290.19	0.1	290.29
28	SUCCESS	2	1	300	0.1	300.1
24	SUCCESS	2	0	308	0.1	308.1
29	SUCCESS	2	0	318	0.1	318.1

existing_MAX_MIN.cloud1 is finished!

BUILD SUCCESSFUL (total time: 9 seconds)

Figure 2: Screenshot of Max_Min

In figure 2 and table 1 shows that the existing Max Min algorithm was implemented which shows the values for different parameters like processing Time ,cost, Waiting time.

Table 1:Parameter Values for Existing Max-Min Algorithm

Total Processing Time	Total Processing Cost	Average Waiting Time
2833.460087082729	849.8950915518895	0.8965732300421054

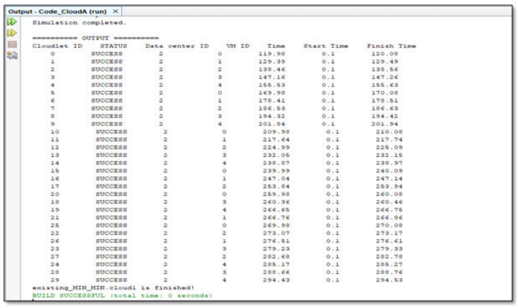


Figure 3: Screenshot of Min-Max

Table 2:Parameter Values for Existing Min-Max Algorithm

Total Processing Time	Total Processing Cost	Average Waiting Time
1412.2865856550648	824.579999999998	1.1279182593589627

In figure 3 and table 2 shows that the existing Max Min algorithm was implemented which shows the values for different parameters like processing Time ,cost, Waiting time.

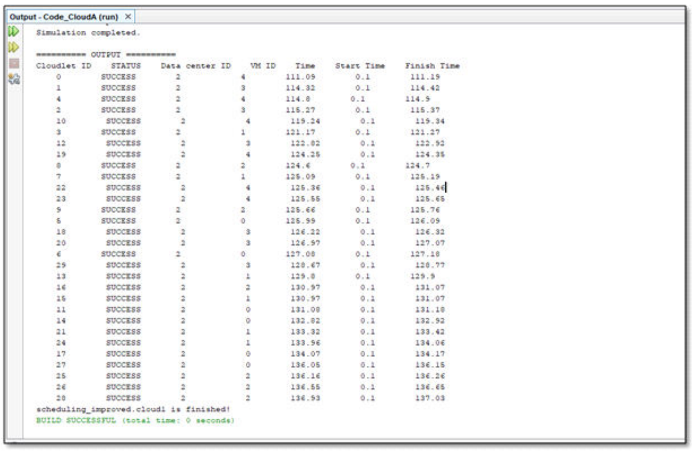


Figure 4:Screenshot of the proposed Work

Table 3:Parameter Values for Existing Min-Max Algorithm

Total Processing Time	Total Processing Cost	Average Waiting Time
661.0794463891023	440.58	0.48778422820697726

In figure 4 ,Proposed task scheduling was accurate based on time, cost, waiting time variables. And the values of the proposed work improves as compared to the existing values.

VII.CONCLUSION

Task scheduling is the process of assigning a specific resource to a task at a specific time. The scheduling process aims to maximize resource efficiency while also reducing waiting times. Task scheduling is among the most challenging jobs in cloud computing. A well-planned schedule can improve effectiveness and stability. The major purpose of this research is to offer an optimal method that combines best features of two most popular Min-Min and Max-Min algorithms. It's worth noting that RALBA methods show that their waiting time is shorter than that of traditional algorithms and that it performs better than the comparable methodologies.

REFERENCES

[1] Peter mell,Timothy Grance “The NIST Definition of Cloud Computing” NIST Special Publication, September 2011.

[2] Anu Gupta “Cloud Computing Growing Interest and Related Concerns” IEEE 2nd International Conference on Computer Technology and Development (ICCTD) ,2010.

[3] Harpreet Kaur,Maninder Singh” Review of Various Scheduling Techniques in Cloud Computing” International Journal of Networking & Parallel Computing, Volume 1, Issue 2, November, 2012.

[4] S.Devipriya, C.Ramesh,” Improved Max-Min Heuristic Model For Task Scheduling In Cloud” IEEE, 2013.

[5] Xiaofang Li, Yingchi Mao, Xianjian Xiao, Yanbin Zhuang,” An Improved Max-Min Task-Scheduling Algorithmfor Elastic Cloud” IEEE International Symposium on Computer, Consumer and Control, 2014.

[6] Saeed Parsa, Reza Entezari-Maleki,” RASA: A New Grid Task Scheduling Algorithm” International Journal of Digital Content Technology and its Applications Volume 3, Number 4, December 2009.

[7] Atul Mishra, Mayanka Katyal “Application of Selective Algorithm for Effective Resource Provisioning In Cloud Computing Environment” International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol. 4, No. 1, February 2014.

[8] Amalarethinam, D. I. G., & Kavitha, S. , “Priority based Performance Improved Algorithm for Meta-task Scheduling in Cloud environment”, 2nd International Conference on Computing and Communications Technologies (ICCCT),2017.

[9] T.Mandal and S.Acharyya, "Optimal Task Scheduling in Cloud Computing Environment : Meta Heuristic Approaches," 2015 Proceedings of International Conference on Electrical Information and Communication Technology (EICT), pp. 24–28, 2015.

[10] Derakhshan, M., & Bateni, Z., "Optimization of tasks in cloud computing based on MAX-MIN, MIN-MIN and priority", 4th International Conference on Web Research (ICWR),2018.

[11] Ibrahim, E., El-Bahnasawy, N. A., & Omara, F. A., "Task Scheduling Algorithm in Cloud Computing Environment Based on Cloud Pricing Models", World Symposium on Computer Applications & Research (WSCAR) , 2016.

[12] Santhosh B., Manjaiah D. H, and L.Pandma Suresh, "A Survey of Various Scheduling Algorithms in Cloud Environment", International Conference on Emerging Technological Trends [ICETT] IEEE, 2016.

[13] Jain, A. and R. Kumari, "An Efficient Resource Utilization Based Integrated Task Scheduling Algorithm", 4th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, 2017.